

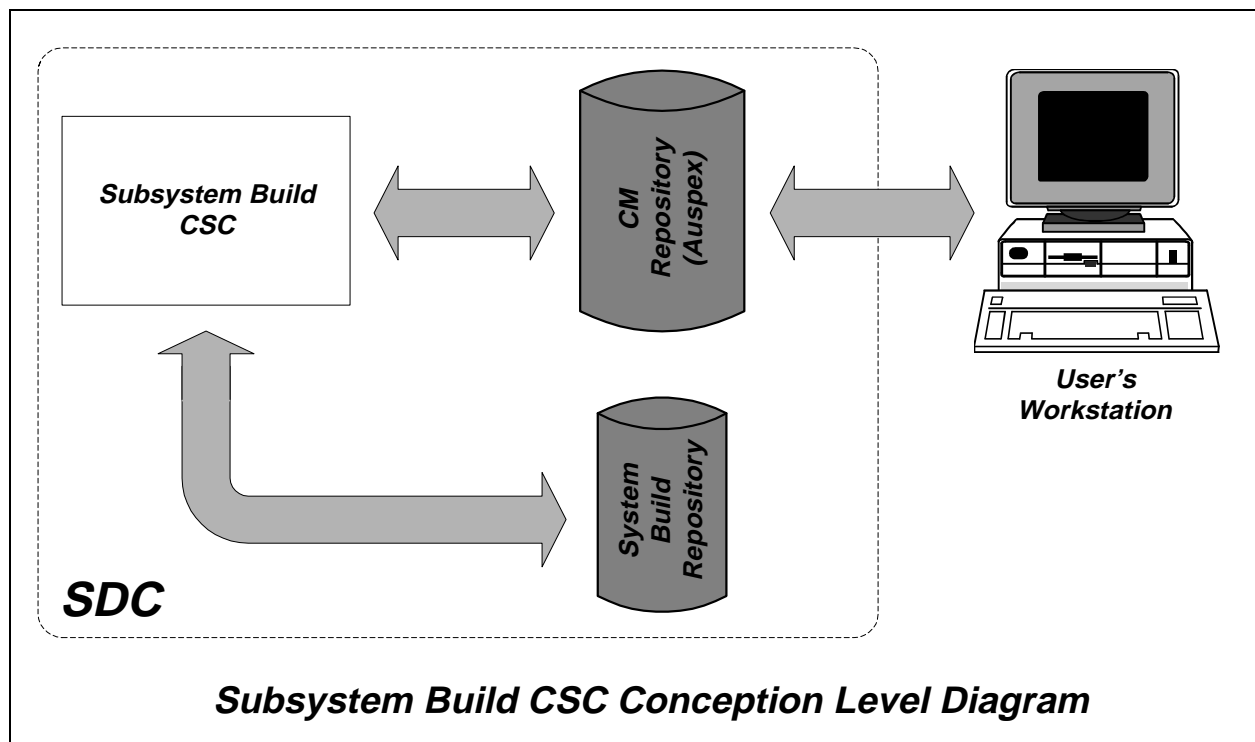
1. Subsystem Build CSC

1.1 Subsystem Build CSC Introduction

1.1.1 Subsystem Build CSC Overview

The Subsystem Build CSC is responsible for defining SCID's, setting up and maintaining the System Build Repository and compiling and linking all system software, including the Gateway software. The System Build Repository will be setup and maintained on the Auspex which is located in the Shuttle Data Center (SDC).

The Subsystem Build CSC also provides the capability to build System Software from the CM Repository into the System Build Repository. This capability is independent of Test builds yet supports multiple Test builds. This capability also provides a method for identifying repeatable System Builds called a System Configuration Identifier (SCID). The SCID, and its revision number, is a repeatable collection of product baselines that support multiple test builds (TCIDs) on a particular set configuration. This set configuration ranges from a set used for Application Debug, to a partially operational set that may be used for test and integration, and finally to a fully operational set.



1.1.2 Subsystem Build CSC Operational Description

The Subsystem Build CSC is an interface between the CM Repository and the System Build Repository. This interface provides a set of integrated tools which allow CLCS users to compile and link the CLCS software. CLCS software is retrieved from the CM Repository, compiled and linked, then stored in the System Build Repository. The directory structure is organized by SCIDs and is further broken down into individual directories for both Development and Controlled baselines.

The Subsystem Build CSC also defines and maintains the SCID Configuration Table. The SCID Configuration Table will track all information pertaining to an SCID. Via a tool, CLCS users will be permitted to name, list, add, modify, or delete both Development and Controlled build products within an SCID. CLCS users will have the capability of placing both tested and untested software in the System Build Repository.

1.2 Subsystem Build CSC Specifications

1.2.1 Subsystem Build CSC Ground Rules

Subsystem Build will operate under the following ground rules:

- The CM Repository will reside on the Auspex which will eventually become part of the SDC.
- The System Build Repository will reside on the Auspex.
- CSCIs to be built must baseline their software in the CM Repository.
- All CSCIs will follow their respective directory formats as established by the System Control CSCI.
- The Subsystem Build CSC shares the System Build Repository with the OPS CM CSC.
- *Italicized* requirements will **not** be implemented in Redstone.
- CSCIs to be built will provide a “make” file for their software pertaining to each platform.
- The term “SCID” as used in this document does not include a Platform Load. A Platform Load is the responsibility of System Services CSCI.
- *Each SCID will be associated with compatible platform loads.*
- A non-baselined Subsystem Load refers to software that has not been baselined in the CM Repository.
- The requirements for the OS Build CSC are being proposed and met by the System Services CSCI.
- CSCIs must follow programming standards established by the Software Development Plan (84K00054).
- CSCIs must follow the standards established by the CM Development Plan (84K00052).

1.2.2 Subsystem Build CSC Functional Requirements

The Functional Requirements area is composed of the following sections:

1. Define SCID.
2. Manage System Build Repository.
3. Build SCID.

1. Define SCID:

- 1.1. The Subsystem Build CSC will provide a means of defining an SCID as a combination of baselined and non-baselined Subsystem Loads and an alphanumeric string with major and minor version numbers.

<SCID name>.<majorID>.<minorID>

Example: ops_scid.1.2

- 1.2. The Subsystem Build CSC will define an SCID to be a combination of one or more of the following Subsystem Loads:
 - HCI
 - DDP
 - CCP
 - DDP/CCP
 - DDP/CCP/HCI
 - Gateway type
- 1.3. The Subsystem Build CSC will provide a means of uniquely naming the baselined products that make up a Subsystem Load within an SCID.

- 1.4. The Subsystem Build CSC will maintain an SCID Configuration Table that contains:
 - a. SCID name.
 - b. SCID version number.
 - c. Creation date/time.
 - d. *A list of subsystem loads, version numbers.*
 - e. SCID status.
 - f. SCID Notes section.
 - g. Target environment.
- 1.5. The Subsystem Build CSC will provide a means of listing the contents of an SCID.
- 1.6. The Subsystem Build CSC will provide a means of adding, deleting, and modifying the contents of an SCID.
- 1.7. The Subsystem Build CSC will provide read access to the SCID Configuration Table.

2. Manage System Build Repository:

- 2.1. The Subsystem Build CSC will prepare the System Build Repository for a new SCID build.
- 2.2. The Subsystem Build CSC will provide a means of deleting an SCID build within the System Build Repository.
- 2.3. The Subsystem Build CSC will provide a means of listing the SCID builds within the System Build Repository.
- 2.4. The Subsystem Build CSC will provide a means of preventing unauthorized updates to the System Build Repository.
- 2.5. *The Subsystem Build CSC will provide the means to detect and prevent the re-creation of an existing SCID.*
- 2.6. The Subsystem Build CSC will provide a means of updating the SCID Configuration Table after an SCID Build.
- 2.7. The Subsystem Build CSC will provide a means of preventing an unauthorized update to the SCID Configuration Table.
- 2.8. The Subsystem Build CSC will provide separate directories for Development and Controlled builds.

3. Build SCID:

- 3.1. The Subsystem Build CSC will provide a means of compiling all of the baselined products that make up the Subsystem Loads within an SCID.
- 3.2. The Subsystem Build CSC will provide a means of reporting compile and link errors and warnings.
- 3.3. The Subsystem Build CSC will capture all compile and link messages from a System Build in a log file.

1.2.3 Subsystem Build CSC Performance Requirements

No specific performance requirements have been established for Redstone.

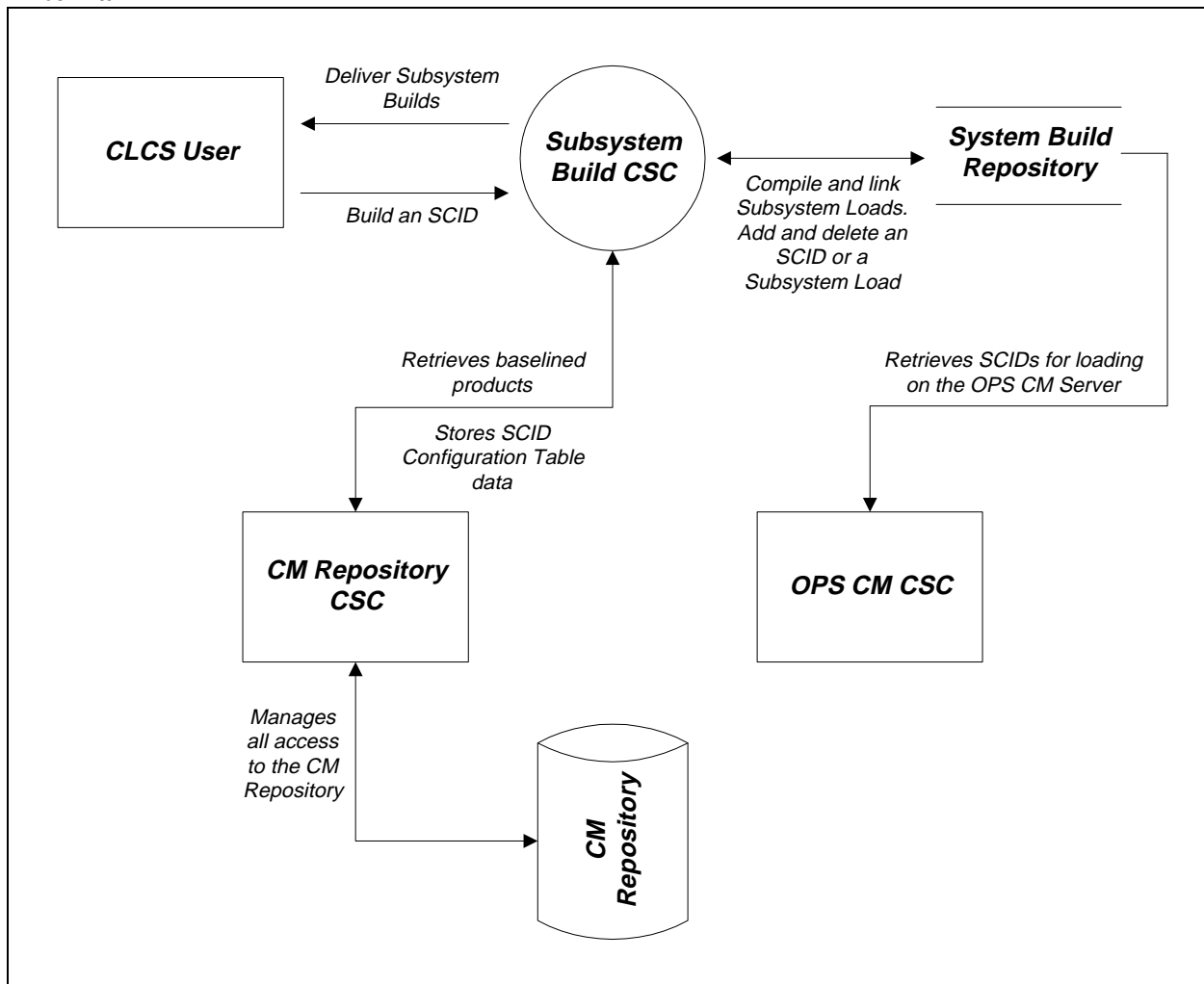
1.2.4 Subsystem Build CSC Interfaces

The Subsystem Build CSC interfaces with the following groups:

- CLCS Users
- OPS CM CSC (*through the shared System Build Repository*)
- CM Repository CSC (*through the CM Repository*)

1.2.5 Subsystem Build CSC Data Flow Diagram

External



System Build CSCI
Subsystem Build CSC
Redstone Requirement Review

May 29, 1997
Version 1.0